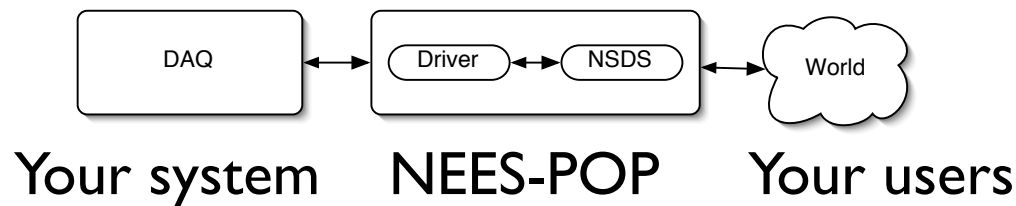
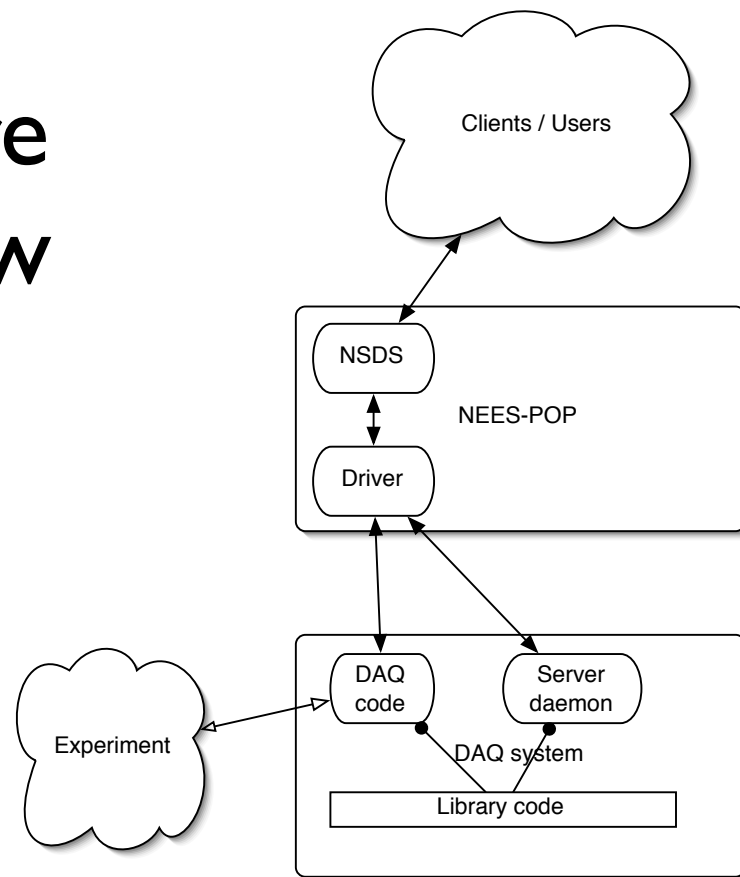


Design of the NEESgrid data acquisition system

Paul Hubbard
April 23, 2003



Architecture and dataflow



Design Principles

- Assume that sites have or will build their own customized DAQ system
- We will not provide a turnkey solution; we do provide a reference implementation in LabVIEW
- Our code should drop into yours with minimal effort: small modules
- Minimize impact on the existing system

Design Principles 2

- DAQ code should not know or notice if network is up or down
 - Auto reconnect logic is our problem, not yours
- Simple protocol that's easy to emulate
 - Code included does so: DI-194 DAQ, fake_daq
- Include examples and testing code

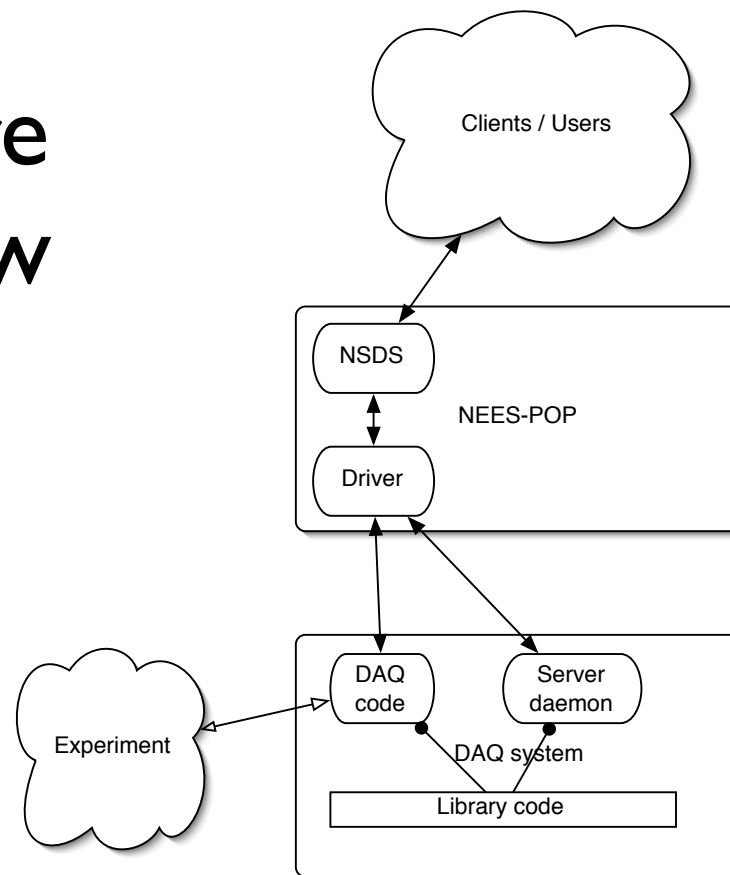
Terminology

- NSDS - NEESgrid Streaming Data Server. Java-based app, runs on the NEES-POP, central to the design.
- NEES-POP - Linux PC, onsite, that runs the NEESgrid collaboration code
- NSDS driver - C code, running on the NEES-POP, to mediate between the NSDS and the DAQ code. Rewrite for other DAQs.

Minimum requirements for running and testing the DAQ code

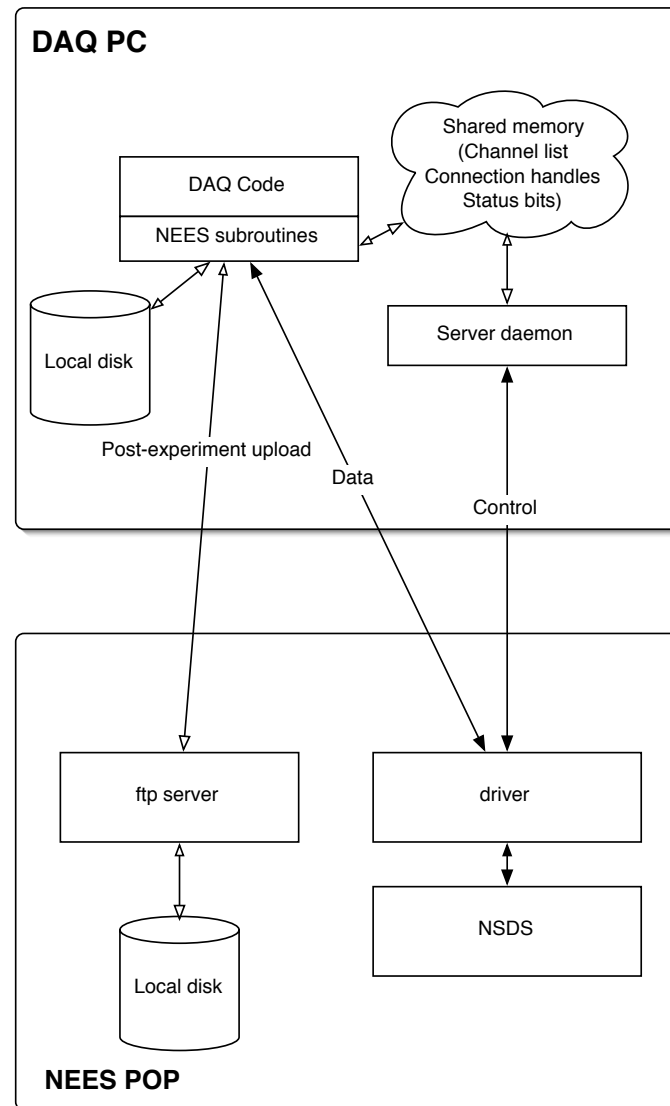
- LabVIEW DAQ system
 - Requires v6.1 and Internet Toolkit (~ \$2,500)
 - Tested on MS Windows (2k and XP). Linux testing in progress: caveats apply to driver support. UEI DAQ, etc.
 - PCI 6036E, DaqPad 6052E, 6070E
- NEESgrid DAQ code NAME!! (see <http://www.mcs.anl.gov/neesgrid/daq-install-instructions.pdf>)
- NEES-POP or other unix box to run the driver code. (<http://www.mcs.anl.gov/neesgrid/driver-install-instructions.pdf>)
- NSDS (on NEES-POP) or simulator (on DAQ PC)

Architecture and dataflow



Normal configuration

- This slide shows the detailed internals on the NEES-POP and DAQ machines.
- Notes
 - Code provided to save to disk and upload via FTP in a standard ASCII format (Tab-delimited with header)
 - Server daemon establishes the data connection but does not touch it otherwise



Example data file

Event ID: 19364841J12
Active channels: ACH1,Temp
Sample rate: 200.000000
Channel units: V,Deg C

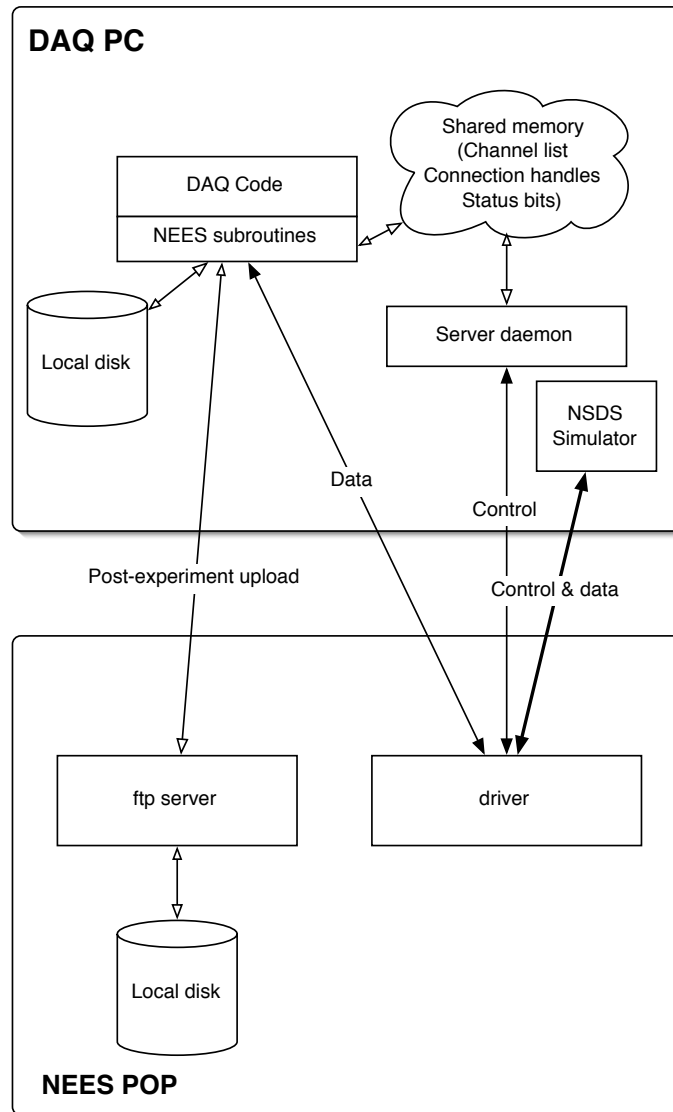
Time	ACH1	Temp
2003-01-24T15:42:02.73399	-0.700531	27.603149
2003-01-24T15:42:02.75000	-1.961975	27.587891
2003-01-24T15:42:02.75500	2.678223	27.618408

Example NSDS stream

2003-01-24T15:42:02.73399	ACH1	-0.700531	Temp	27.603149
2003-01-24T15:42:02.75000	ACH1	-1.961975	Temp	27.587891
2003-01-24T15:42:02.75500	ACH1	2.678223	Temp	27.618408

Testing with the NSDS simulator

- Same design, but now NSDS is simulated in LabVIEW on the DAQ PC.
- Verify the driver and network
- No need to have CHEF available to see data plots
- Code supplied includes:
 - Simple single-channel subscribe and plot
 - Stress test - subscribe to all, no plotting
 - Subscribe and plot all
- This is how I normally develop
-



Misc Notes

- Separate TCP connections for control and data (ports 55055 & 55056 by default)
- Control is bidirectional, data is unidirectional (DAQ -> world)
- Driver <-> NSDS connection on 42420/I
- Commands are sent by the NSDS and answered by 'server daemon' on the DAQ PC.
- Data streams from the DAQ, via the driver, thence to the NSDS

Driver details

- Main function is to *initiate* and maintain connections
- Main loop
 - Connect to DAQ
 - Connect to NSDS
 - Forward on both channels until a TCP error occurs
 - Responds to NSDS's initial ID request; ID is configurable via the command line
- POSIX C, pthreads, runs on most Unix variants
- Source code is heavily Doxygen-documented; see the html subdirectory for the results.

DAQ on other platforms

- Not everyone runs LV for DAQ
- Read 'NSDS-Driver Protocol' document and decide if you want to use it or rewrite the driver (<http://www.mcs.anl.gov/neesgrid/>)
- Example source code in C:
 - fake_daq (in nsds-driver package)
 - neesdaq (DI-194, in nees-di package)
- All source code has Doxygen docs included

Where To Go From Here

- <http://www.mcs.anl.gov/neesgrid>
 - Pointers to CVS, mailing list, MUD
 - PDF docs
 - Bugzilla
- <http://www.ni.com/labview>
- phubbard@anl.gov